UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/693,589	10/24/2003	Jeffrey P. Snover	MS1-1742US	1106

22801 7590 04/05/2006

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
SINGH, RACHNA	
ART UNIT	**PAPER NUMBER**
2176	

DATE MAILED: 04/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/693,589	SNOVER ET AL.	
	Examiner	Art Unit	
	Rachna Singh	2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 January 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: Arguments submitted on 01/16/06.

2. Claims 1-37 are pending. Claims 1, 12, 20, and 26 are independent claims.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

4. Claims 1-37 are rejected under 35 U.S.C. 102(e) as being anticipated by Muhlestein et al., US 2003/0018765 A1 (field 01/23/03).

In reference to claim 1, Muhlestein teaches a system and method for accessing management functionality through a command line utility. See page 2, paragraphs [0011]-[0015] and page 34, first column. Muhlestein discloses an object model command schema defining a mapping between one or more commands and an object model target schema, the one or more commands generated by the command schema and configured to operate against the target schema through the command line utility. The command schema is extensible to permit the generation of additional commands to be added to the one or more commands. On page 2, paragraph [0012], Muhlestein

states "a set of commands for the WMI command line utility is configured by an underlying object model command schema that defines a mapping between the commands and the WMI schema". A pipeline is used to hold several commands in a command line simultaneously thus when Muhlestein states on page 2, paragraph [0012], "***a set of commands*** for the WMI command line utility is configured by an underlying object model command schema that defines a mapping between the commands and the WMI schema, one command can be entered into the command line", he is describing a pipeline. Compare to ***"in an operating environment supporting a subsequent command within the pipeline being configured to communicate with a prior command within the pipeline through a parseable object emitted from the prior command, the operating environment configured to support the execution of commands within the same process."*** The command schema comprises an alias class defining a command template and a format class as a subclass to the alias class, each instance of the format class representing a list of properties that will be returned through processing an alias. The alias object is a command that is executed in order to capture the features of the target class and to facilitate a task. Alias objects are instances of command-related classes organized into a command schema. The command schema defines the commands and the command line utility uses the aliases to interpret command information and apply the interpretation against the target schema. Properties of the alias object can be arranged in named formats that include property values that can be formatted for display. See page 5. The formats attribute includes a list of properties that are to be displayed using the given

format. See page 34 and page 6, paragraphs [0054]-[0059]. For example, a command is received through an interface by the WMI command line utility. The utility interprets the command based on its definition. The command is executed as a series of API calls against a target namespace. The WMI infrastructure at the target station then performs the WMI operations against the target object. The WMI data retrieved is transformed at operation into XML information that is readable by the command line utility. See page 9, paragraph [0091]-page 10, paragraph [0095]. Compare to ***"receiving the parseable object emitted from the prior command, the parseable object having at least one method; obtaining a data type for the parseable object; obtaining format information describing a format for the data type;"*** Muhlestein further teaches permitting the generation of additional commands to be added to one or more commands. The user can parameterize commands by specifying locations of command definitions and target system objects. A connection class, a subclass to the alias class, defines the connection instances, each connection instance representing connection parameters used by an alias to establish a connection to target namespace within the target schema. The format class represents a list of properties that will be returned through processing an alias. The command schema is extensible to permit the generation of additional commands to be added to the set of commands. See page 8 and page 34. Compare to ***"emitting a format object for access by another subsequent command, the format object being based on the format information"***.

In reference to claim 2, Muhlestein teaches the command line utility reads XML information. See page 10, paragraph [0095].

In reference to claim 3, Muhlestein teaches the command schema comprises an alias class defining a command template and a format class as a subclass to the alias class, each instance of the format class representing a list of properties that will be returned through processing an alias. The alias object is a command that is executed in order to capture the features of the target class and to facilitate a task. Alias objects are instances of command-related classes organized into a command schema. Properties of the alias object can be arranged in named formats that include property values that can be formatted for display. See page 5. The formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059].

In reference to claim 7, Muhlestein teaches the command schema comprises an alias class defining a command template and a format class as a subclass to the alias class, each instance of the format class representing a list of properties that will be returned through processing an alias. The alias object is a command that is executed in order to capture the features of the target class and to facilitate a task. Alias objects are instances of command-related classes organized into a command schema. The command schema defines the commands and the command line utility uses the aliases to interpret command information and apply the interpretation against the target schema. Properties of the alias object can be arranged in named formats that include property values that can be formatted for display. See page 5. The formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059]. Muhlestein further teaches permitting the

generation of additional commands to be added to one or more commands. The user can parameterize commands by specifying locations of command definitions and target system objects. A connection class, a subclass to the alias class, defines the connection instances, each connection instance representing connection parameters used by an alias to establish a connection to target namespace within the target schema. The format class represents a list of properties that will be returned through processing an alias. The command schema is extensible to permit the generation of additional commands to be added to the set of commands. See page 8 and page 34.

In reference to claims 8-10, Muhlestein teaches the formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059]. Muhlestein further teaches the Format string provides a location of an XSL file that can be used to format the display for the list of properties being returned to the user through the alias. The WMI command line utility returns information as XML documents which are processed using XSL to render reports for the user in multiple formats as determined by the XSL file located through the format string. Any display strategies to be used in displaying properties will be defined by the XSL style sheet in formatting displays. See page 6, paragraph [0063].

In reference to claim 11, Muhlestein teaches properties of the alias object can be arranged in named formats that include property values that can be formatted for display. See page 5. The formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059].

Claims 12-19 are rejected under the same rationale used in claims 1-3 and 7-11 respectively above.

Claims 20-25 are rejected under the same rationale used in claims 1-2, 11, 7-8, and 10 respectively above.

In reference to claims 26, Muhlestein teaches a command line utility comprising an object model command schema to define a mapping between one or more commands an object model target schema, the one or more commands generated by the command schema and configured to operate against the target schema through the command line. Compare to ***"a method for providing a data driven command line output"***. The command line utility comprises an alias class defining a command template, each alias of the alias class representing a single command and a format class as a subclass to the alias class, each instance of the format class representing a list of properties that will be returned through processing an alias. The command schema defines the commands and the command line utility uses the aliases to interpret command information and apply the interpretation against the target schema. See page 2, page 6, paragraphs [0054]-[0059], and page 34. For example, a command is received through an interface by the WMI command line utility. The utility interprets the command based on its definition. The command is executed as a series of API calls against a target namespace. The WMI infrastructure at the target station then performs the WMI operations against the target object. The WMI data retrieved is transformed at operation into XML information that is readable by the command line utility. See page 9, paragraph [0091]-page 10, paragraph [0095]. Compare to ***"receiving command-***

line instruction containing an output command configured to receive at least one object, the object having at least one method; and executing the output command to manipulate at least one object". Properties of the alias object can be arranged in named formats that include property values that can be formatted for display. See page 5. The formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059]. Compare to ***"output a result to an output destination"***.

In reference to claims 27-28 Muhlestein teaches an object-based command-line environment. See abstract and pages 1-2.

In reference to claims 32, 36, and 37, Muhlestein teaches the formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059]. Muhlestein further teaches the Format string provides a location of an XSL file that can be used to format the display for the list of properties being returned to the user through the alias. The WMI command line utility returns information as XML documents which are processed using XSL to render reports for the user in multiple formats as determined by the XSL file located through the format string. Any display strategies to be used in displaying properties will be defined by the XSL style sheet in formatting displays. See page 6, paragraph [0063].

In reference to claims 33-35, Muhlestein further teaches permitting the generation of additional commands to be added to one or more commands. The user can parameterize commands by specifying locations of command definitions and target system objects. A connection class, a subclass to the alias class, defines the

connection instances, each connection instance representing connection parameters used by an alias to establish a connection to target namespace within the target schema. The format class represents a list of properties that will be returned through processing an alias. The command schema is extensible to permit the generation of additional commands to be added to the set of commands. See page 8 and page 34

In reference to claims 4-6 and 29-31, Muhlestein teaches the formats attribute includes a list of properties that are to be displayed using the given format. See page 34 and page 6, paragraphs [0054]-[0059]. Muhlestein further teaches the Format string provides a location of an XSL file that can be used to format the display for the list of properties being returned to the user through the alias. The WMI command line utility returns information as XML documents which are processed using XSL to render reports for the user in multiple formats as determined by the XSL file located through the format string. Any display strategies to be used in displaying properties will be defined by the XSL style sheet in formatting displays. See page 6, paragraph [0063]. Muhlestein further discloses displaying the formatted data through a graphical user interface. See page 36, claims 34-36 and figures 2 (illustrates a console) and figure 3.

Response to Arguments

5. Applicant's arguments filed 01/16/06 have been fully considered but they are not persuasive.

With respect to claims 1, 12, and 20, Applicant argues Muhlestein does not teach pipelining commands. Examiner respectfully disagrees. Muhlestein teaches a

command line in which *a set of commands* for a WMI command line utility is entered on the command line. See page 2, paragraph [0012] where Muhlestein states "a set of commands for the WMI command line utility is configured by an underlying object model command schema that defines a mapping between the commands and the WMI schema".

Applicant further argues only a single WMIC command can be entered on the command line. Examiner respectfully disagrees with Applicant's assertion. On page 2, paragraph [0012], Muhlestein states "***a set of commands*** for the WMI command line utility is configured by an underlying object model command schema that defines a mapping between the commands and the WMI schema". Thus more than one command can be entered.

In view of the same arguments, Applicant asserts Muhlestein does not teach a subsequent command within the pipeline being configured to communicate with a prior command within the pipeline through a parseable object emitted from the prior command. Examiner respectfully disagrees. Applicant states a parseable object is defined as input in which properties and values may be discerned. Applicant admits the Muhlestein reference teaches "parseable object" on page 14, lines 5-9 of their response, "even though the Muhlestein reference discusses WMI input and the present application discusses that ***one type of parseable object may be Windows Management Instrumentation input***. . .". Furthermore, on page 2, paragraph [0012], Muhlestein states "***a set of commands*** for the WMI command line utility is configured by an underlying object model command schema that defines a mapping between the

commands and the WMI schema". Thus more than one command can be entered into the command line.

Applicant argues Muhlestein does not teach, "each command communicate with other commands through a parseable object via a pipeline". A pipeline is used to hold several commands in a command line simultaneously thus when Muhlestein states on page 2, paragraph [0012], "***a set of commands*** for the WMI command line utility is configured by an underlying object model command schema that defines a mapping between the commands and the WMI schema, one command can be entered into the command line. Compare to ***"pipeline"***.

Applicant argues Muhlestein does not teach "emitting a format object for access by another subsequent command, the format object being based on the format information". Examiner respectfully disagrees. Muhlestein teaches permitting the generation of additional commands to be added to one or more commands. The user can parameterize commands by specifying locations of command definitions and target system objects. A connection class, a subclass to the alias class, defines the connection instances, each connection instance representing connection parameters used by an alias to establish a connection to target namespace within the target schema. The format class represents a list of properties that will be returned through processing an alias. The command schema is extensible to permit the generation of additional commands to be added to the set of commands. See page 8 and page 34.

With respect to claim 26, Applicant argues Muhlestein does not teach a command receiving an object or how an object is received could have at least one

method. Examiner respectfully disagrees. Muhlestein teaches a command line utility comprising an object model command schema to define a mapping between one or more commands an object model target schema, the one or more commands generated by the command schema and configured to operate against the target schema through the command line. The command line utility comprises an alias class defining a command template, each alias of the alias class representing a single command and a format class as a subclass to the alias class, each instance of the format class representing a list of properties that will be returned through processing an alias. The command schema defines the commands and the command line utility uses the aliases to interpret command information and apply the interpretation against the target schema. See page 2, page 6, paragraphs [0054]-[0059], and page 34. For example, a command is received through an interface by the WMI command line utility. The utility interprets the command based on its alias definition. The command is executed as a series of API calls against a target namespace. The WMI infrastructure at the target station then performs the WMI operations against the target object. The WMI data retrieved is transformed at operation into XML information that is readable by the command line utility. See page 9, paragraph [0091]-page 10, paragraph [0095].

Applicant challenges Examiner's statement with regards to claims 4-6 and 29-31. Upon further review, Examiner believes the Muhlestein reference expressly teaches these features and thus withdraws the previous 103 rejection and now rejects claims 4-6 and 29-31 under 35 U.S.C. 102, as rejected above.

In view of the comments above, the rejections are maintained.

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Rachna Singh whose telephone number is 571-272-4099. The examiner can normally be reached on M-F (8:30AM-6:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on 571-272-4136. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

RS
03/30/06

Doug Hutton
Primary Examiner
Art Unit 2176